2B-5

IBM 3179-G MICROCODE TASK CONTROL MECHANISM

Tadashi MIZUTANI, Tsuyoshi INOMATA
Yamato Laboratory - IBM Japan

INTRODUCTION

3179-G is a color graphics keyboard display terminal which attaches to 3274 control unit. It can support mouse, plotter and screen printer provided with expansion unit.

A large size of microcode is mounted in 3179-G with INTEL 8088 MPU. It is required that the best requested task at the moment is executed in order to achieve high performance. A supervisor called task manager is implemented for this purpose which controls 14 main modules as task.

TASK MANAGER AND TASK

Task manager is a interrupt handler. Two kinds of interrupt, external interrupt (EXI) and non-maskable interrupt (NMI) occur in 3179-G. Whenever interrupt happens, control is passed to task manager. Various tasks are requested by these interrupts. Task manager analyses the interrupt and determines which task should be executed.

Task manager invokes each task by software interrupt (INT). Each task returns control by interrupt return (INTRET). Also, each task may request the other task execution. In which case task manager is invoked by INT. And when the task which is suspended is waked up again, task manager does by INTRET. Fig. 1 shows the invocation of task manager and each task.

TASK CONTROL MATRIX

Task manager controls task execution by introducing fixed priority among tasks. Whenever task manager is invoked, it passes control to the highest priority task which has been requested.

For this control, task manager uses Task Control Matrix (TCM). Fig. 2 shows the example of TCM at some moment.

- BUSY bit shows the currently running task. Only one "ON" bit can exist in BUSY column.
- WAIT bit indicates task which has been invoked once but is now suspended for higher priority task execution. At most one task can be made waiting for each task.
- READY is used as a counter. It shows the number of taks which are ready for execution in queue. Queued execution requests are served in FIFO order for each task.

Task manager increments corresponding READY counter when there arise any requests. Then decision is made from following two alternatives.

Alternative 1: Generate new task by INT.

- READY --- BUSY

  Decrement the highest READY counter and set BUSY bit "ON" for it. The old BUSY bit is removed when previously running task has completed, or moved to WAIT bit when it has not yet.

IBM 3179-G MICROCODE TASK CONTROL MECHANISM

Tadashi MIZUTANI, Tsuyoshi INOMATA
YAMATO Laboratory - IBM Japan

Alternative 2: Invoke the suspended task by INTRET.

. BUSY --- BUSY

Continue the execution of currently running task.

. WAIT --- BUSY

Return the control back to the task which is suspended. Move the highest WAIT bit to BUSY bit. The old BUSY bit is removed since previously running task has completed.

Consequently, TCM is always kept to satisfy that no higher WAIT bit or READY counter exist than the BUSY bit.

CHARACTERISTICS

Owing to fixed priority, the control mechanism is rather simple. Task manager uses only TCM to select control. This simplicity or safety is quite an important factor for supervisor. As a result, 3179-G has realized high performance and reliability.

However, weakness of fixed priority does exist in other hands. First, higher task can not wait for lower task execution. Priority order is defined to permit these requests as much as possible. Second, lower task can be made waiting for a long time. Some kind of mechanism, for example time sharing, is needed to provide chance to run for lower tasks. But improvement for these items are quite restricted under the current stack operation.

Priority is controlled only by task manager. Each task does not have to take care about priority. In case it is necessary to change priority, only task manager is needed to be modified. In fact, task manager is designed flexible for changing priority. But expandability is rather poor. That is, adding or removing a task is not easy.

CONCLUSION

Mechanism of task control which has been implemented using INTEL 8088 MPU is described in this paper. By introducing fixed priority among tasks, simple and powerful task control is achieved. To add more function of task manager, especially time sharing mechanism, will be the future improvement item.

Anyway, it is no doubt that the mechanism here is a basic one for implementing task control mechanism with INTEL 80286/80386.

FIGURES

FIG. 1. INVOCATION OF TASK MANAGER AND TASK

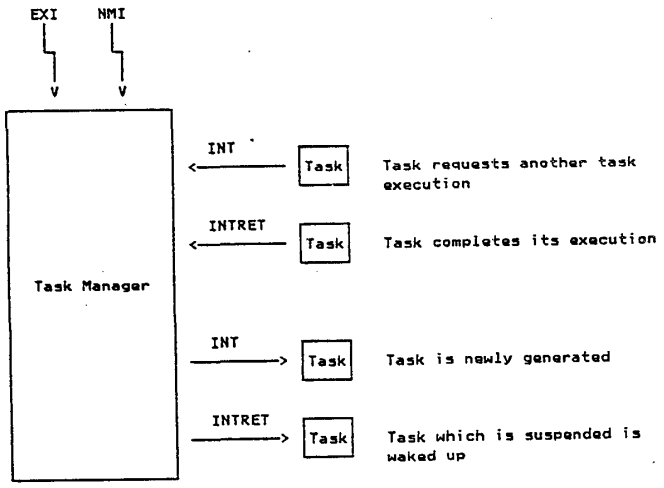

FIG. 2. TASK CONTROL MATRIX

Priority	Task	BUSY	WAIT	READY
1	ERR			
2	TCA			
3	DMP			
4	IND	1		
5	MOU			
6	KTK		1	2
7	DSL			1
8	OTS			
9	MAN		1	
10	SPR			
11	PLT		1	2
12	CTS			
13	POR			
14	BGD		1	